Pattern Recognition 145 (2024) 109974

Contents lists available at ScienceDirect

Pattern Recognition

journal homepage: www.elsevier.com/locate/pr

Source-free domain adaptation with Class Prototype Discovery

Lihua Zhou [a,1], Nianxin Li [a,1], Mao Ye [a,*], Xiatian Zhu [b], Song Tang [c]

[a] School of Computer Science and Engineering, University of Electronic Science and Technology of China, Chengdu 611731, China
[b] Surrey Institute for People-Centred Artificial Intelligence, CVSSP, University of Surrey, Guildford, United Kingdom
[c] Institute of Machine Intelligence (IMI), University of Shanghai for Science and Technology, Shanghai, China

ARTICLE INFO

Keywords:
Source-free domain adaptation
Class prototype discovery
Pseudo-labels
Prototype regularization

ABSTRACT

Source-free domain adaptation requires no access to the source domain training data during unsupervised domain adaption. This is critical for meeting particular data sharing, privacy, and license constraints, whilst raising novel algorithmic challenges. Existing source-free domain adaptation methods rely on either generating pseudo samples/prototypes of source or target domain style, or simply leveraging pseudo-labels (self-training). They suffer from low-quality generated samples/prototypes or noisy pseudo-label target samples. In this work, we address both limitations by introducing a novel **Class Prototype Discovery** (CPD) method. In contrast to all alternatives, our CPD is established on a set of semantic class prototypes each constructed for representing a specific class. By designing a classification score based prototype learning mechanism, we reformulate the source-free domain adaptation problem to class prototype optimization using all the target domain training data, and without the need for data generation. Then, class prototypes are used to cluster target features to assign them pseudo-labels, which highly complements the conventional self-training strategy. Besides, a prototype regularization is introduced for exploiting well-established distribution alignment based on pseudo-labeled target samples and class prototypes. Along with theoretical analysis, we conduct extensive experiments on three standard benchmarks to validate the performance advantages of our CPD over the state-of-the-art models.

1. Introduction

Whilst supervised deep learning has made remarkable progress in computer vision [1], but this breakthrough requires a large amount of labeled data. In many real-world scenarios such as autonomous driving and medical imaging, it is difficult to obtain a large amount of labeled data per domain for enabling supervised model optimization. Typically, each domain has its own specific characteristics, such as illuminations, colors, and backgrounds, giving rise to the challenging domain shift. Specifically, a model trained on one domain (e.g., the source domain) in a supervised learning manner often suffers poor generalization to a new domain (e.g., the target domain). To avoid exhaustive per-domain training data labeling, domain adaptation [2–4] is a viable solution, which aims to transfer the knowledge learned from a labeled source domain to an unlabeled target domain to help model perform better in target domain.

Most existing domain adaptation methods focus on unsupervised domain adaptation (UDA), which assume the simultaneous access to both labeled source domain training data and unlabeled target domain training data during model adaptation [2]. With the increasingly strict regularization imposed on data privacy and license, the research community has started to take action in innovation in order to fulfill the legal responsibility and obligation. This has recently led to an emerging research attention on source-free domain adaptation (SFDA) [5,6]. Building upon UDA, it further enforces a constraint of *no access to the source domain training data*, with only the pre-trained model exposed to domain adaptation.

There have been a number of SFDA works introduced in two lines. The first line [6–8] takes the distribution alignment idea, largely following the most common paradigm of conventional UDA models. To that end, they often require to generate pseudo training samples or class prototypes in source or target style for distribution alignment. This generation task itself however is sufficiently challenging and remains open and largely unsolved. Poor quality samples would significantly harm the model adaption process and lead to inferior model generalization on the target domain (Fig. 1(a)). The second line of methods [5] instead adopt a self-training strategy without data generation. Nonetheless, this approach has fundamental defects: (1) Failing to fully exploit the target domain training data as only a fraction of high-confidence samples can

* Corresponding author.
E-mail address: cvlab.uestc@gmail.com (M. Ye).
[1] Equal contribution.

https://doi.org/10.1016/j.patcog.2023.109974
Received 1 April 2023; Received in revised form 13 July 2023; Accepted 12 September 2023
Available online 19 September 2023

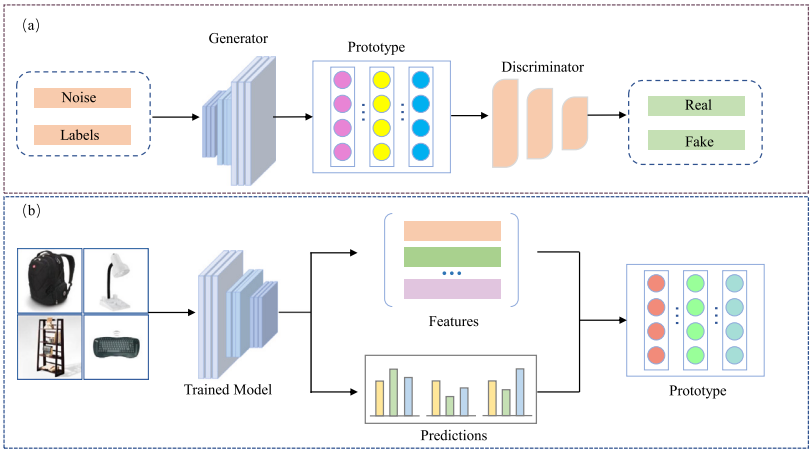

Fig. 1. Illustration of (a) the traditional distribution alignment methods *vs.* (b) our class prototype discovery (CPD) method. Traditional distribution alignment methods generate the class prototypes (or images) to perform domain adaptation, which suffers from the pitfalls of generating tasks. While our method constructs class prototypes by designing a classification score based prototype learning mechanism.

be used in order to minimize false pseudo-labels; (2) Time-consuming or space-consuming in order to obtain some additional supervision information (*e.g.* pseudo-label). (3) Lacking a domain alignment scheme that has been well established and verified effective in UDA.

To address the above two limitations of existing SFDA methods, in this work we propose a novel **Class Prototype Discovery** (CPD) method. Unlike the previous distribution alignment methods that need to generate pseudo samples or class prototypes, we create a set of semantic class prototypes (each for one class) by designing a classification score based learning mechanism to leverage all the target domain samples. The SFDA problem is hence turned to a class prototype discovery process. During training, we use the current batch of target samples to incrementally update the class prototypes with the exponential moving average (EMA) strategy. To fully capitalize the prototypes, we further integrate self-training and distribution alignment. Concretely, with class prototypes as the initial cluster centers, we perform clustering to generate pseudo-labels.

We make the following **contributions**. **(1)** We propose a novel *Class Prototype Discovery* (CPD) method for solving the SFDA problem, without the need for generating source-domain like training data. Making a full use of all the training data of target domain, it can extract richer class discriminative information for more effective domain adaptation. This goes beyond the previous self-training alternative methods that need more complexity to obtain extra supervision information. **(2)** A prototype regularization is further introduced based on the pseudo-label and distribution alignment strategies. **(3)** Despite its simple design, extensive experiments on three standard image recognition benchmarks demonstrate that our CPD outperforms a wide variety of state-of-the-art methods for both UDA and SFDA, often by a large margin.

2. Related work

Unsupervised domain adaptation (UDA) aims to transfer knowledge learned from a labeled source domain to an unlabeled target domain, which assumes the simultaneous access to both labeled source domain training data and unlabeled target domain training data during model adaptation. The existing UDA methods can be roughly divided into seven categories. The first category is *statistic moment matching*, with the main idea as defining and minimizing some loss function on distribution discrepancy [9–11]. PAS [9] progressively anchor the target samples which have relatively reliable pseudo labels for adaptation. DCAN [10] aligns the conditional distributions by minimizing Conditional Maximum Mean Discrepancy, and extracts discriminant information from the target domain by maximizing the mutual information between samples and the prediction labels. CIDA [11] exploits the mutual information to measure the independence/dependence between features and the corresponding class/domain labels to learn a discriminative and domain-adaptive representation for both domains. The second category uses the *adversarial learning* with a domain discriminator for domain discrimination [2,12,13]. For domain alignment, DANN [12] and CDAN [2] forces the feature network to produce the representations such that the domain discriminator fails to tell their domain . CRL [13] designs a novel class restriction loss by the intra-class centralization and inter-class normalization to alleviate the noisy pseudo label overfitting problem. The third category leverages the *adversarial generalization* that combines the domain discriminator with a generator [14]. CoGAN [14] generates fake data and aligns the distribution between the two domains at the pixel level. The fourth category applies the *bi-classifier adversarial learning* [15,16]. MCD [15] plays a minimax game with a single feature extractor and two distinct classifiers during adaptation. It maximizes the prediction discrepancy when training the classifiers and minimizes the prediction discrepancy when training the feature extractor. While CDAL [16] further solve the ambiguous target samples by proposed ECI strategy. The fifth category is *Optimal transport* [17]. DeepJDOT [17] consists of two steps: Finding a coupling matrix for connecting each source sample and target sample, followed by minimizing the cost of these pair-wise connections. The sixth category resorts to *self-supervised learning* (SSL). It usually incorporates an auxiliary SSL task for further minimizing the cross-domain discrepancy [18,19]. MNCP [18] proposes a prototype-guided domain-invariant feature representation method to avoid transferring harmful knowledge from the source domain to the target domain. DCJAN [19] constructs a multitask pipeline by integrating jigsaw puzzle based self-training and conditional constraint in the autoencoder framework. The final category adopts the *ensemble learning strategy*. A typical overview pf adopted is the Mean Teacher overview of with a student network and a teacher network involved [20,21].

Source-free domain adaptation (SFDA) is a new, special case of unsupervised domain adaptation. It eliminates the access to the source domain training data. This is required in many application cases with constraints in data privacy, decentralized computing and computational resources. We investigate the SFDA setting in this paper. The existing SFDA methods can be divided into two groups. The first group [6–8] adopts the distribution alignment idea following traditional UDA models. In particular, SFIT [7] leverages the batch-norm layers of a pre-trained source domain model to stylize the training data of target domain into the source domain. Given pseudo-source

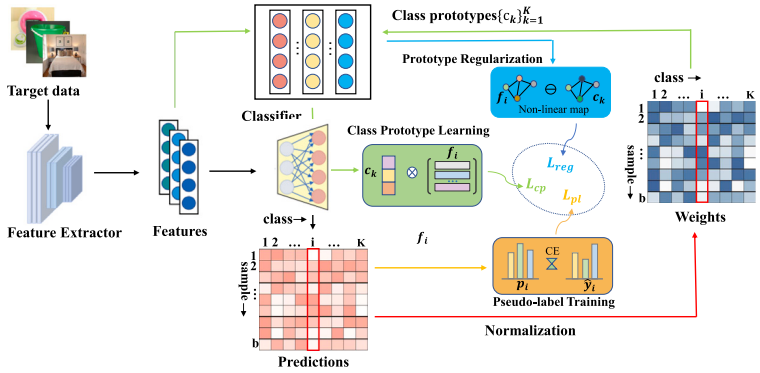

Fig. 2. Overview of the proposed *Class Prototype Discovery* (CPD) method. In SFDA, the model takes as input only the unlabeled training data of target domain. The core of CPD is to discover **(a)** a set of class prototypes $\{c_k\}_{k=1}^K$ that can drive the model adaptation to fit the characteristics of target domain (green arrow). Each class prototype c_k is designated for a specific class $k \in \{1, \dots, K\}$, residing in the feature space. They are updated progressively during each mini-batch training (Eq. (2)), conditioned on **(b)** the weight constraints derived from the classification scores (red arrow). We further introduce **(c)** a prototype regularization for conditional distribution alignment across domains, based on the class prototypes $\{c_k\}_{k=1}^K$ (blue arrow). **(d)** Finally, we also integrate the self-training strategy by pseudo-label estimation at the mini-batch level (orange arrow).

domain data, existing UDA methods can be then applied. In contrast, 3C-GAN [6] instead designs a generator to produce target-domain-style training samples subject to the classifier of the model pre-trained on the source domain, followed by applying a clustering-based regularization and weight regularization to better generalize the source-domain model. On the other hand, SDDA [8] applies conditional GAN framework to generate labeled samples, and aligns the distribution between two domains at the feature level and the pixel level. SFDA-DE [22] uses class prototypes learned by pre-trained classifier as the initialization of feature center for each class and clusters target features with spherical k-means. A common challenge of these methods is in generating high-quality training data or class prototypes which remain difficult and unsatisfactory.

The second group of methods adopts a self-training learning strategy. The main idea is to apply additional supervision information for model updating. For example, SHOT [5] adopts the information maximization principle to make the prediction outputs scatted in the target domain. D-MCD [23] devises a strong–weak self-training paradigm to reduce the label noise in the high-confidence pseudo samples. PSFDA faster R-CNN [24] devises a scheme of iteratively updating global class prototype and a more accurate pseudo-label generation method combining semantic information and image information. Whilst achieving good performance, these methods are fundamentally limited in leveraging the useful information of low-confidence samples or assuming class balanced training data [5]. To overcome this problem, we propose the idea of class prototype discovery in this work wherein all the training samples of target domain can be exploited to support the model adaptation in a confidence-aware manner.

3. Method

3.1. Problem statement

In source-free domain adaptation (SFDA), we have access to a model $\boldsymbol{\Theta}_s$ pre-trained on a labeled dataset $D_s = \{(x_i^s, y_i^s)\}_{i=1}^{n_s}$ from a source domain, where x_i^s denotes the ith image with the class label $y_i \in \{1, 2, \dots, K\}$ and of K a total classes. Given a target domain $\mathcal{T}$ represented by an unlabeled dataset $D_t = \{(x_i)\}_{i=1}^{n}$, we aim to adapt the source model $\boldsymbol{\Theta}_s$ to be better performing without access to the source training data D_s. Often, there exists distribution shift across domains that challenges the model $\boldsymbol{\Theta}_s$.

Overview: Given an image x, the model aims to classify it into one of the K classes accurately. As shown in Fig. 2, we adopt the common CNN architecture [1]: (1) first extracting a feature vector $\boldsymbol{f} = g(x) \in \mathbb{R}^d$ of an image x with the feature dimension d, and (2) passing through a classifier $h(\cdot)$ to obtain the classification score $\boldsymbol{p} = h(\boldsymbol{f}) \in \mathbb{R}^K$.

The core contribution of this work is a new ***Class Prototype Discovery*** (CPD) strategy. A class prototype is a vector representation of a specific class in the feature space $\boldsymbol{f}$. CPD aims to discover an optimal prototype $\boldsymbol{c}_k$ for each class k with unlabeled target domain data D_t in the sense that they can well capture the semantic characteristics of target domain, therefore realizing domain adaptation. Crucially, our CPD enables to simplify the model architecture because each individual class prototype can be easily evaluated by the classifier. Specifically, to evaluate each class prototype $\boldsymbol{c}_k$ during domain adaptation, we merely need to apply the classifier and match the classification score $\boldsymbol{p}_k = h(\boldsymbol{c}_k)$ against the designated class label k. This can be realized by the cross-entropy based classification optimization.

3.2. Class prototype discovery

Our CPD is designed to be compatible with mini-batch based deep learning. Given a mini-batch target domain training data $\{x_i\}_{i=1}^B$ at each training iteration, we obtain the feature vectors $\{\boldsymbol{f}_i\}_{i=1}^B$ and classification scores $\{\boldsymbol{p}_i\}_{i=1}^B$. The key challenge is to update the class prototypes $\{\boldsymbol{c}_k\}_{k=1}^K$ with these unlabeled training data.

At the absence of class label, we propose to accumulate the visual knowledge from the unlabeled data to the class prototypes in a confidence-aware manner. Specifically, we start by introducing a per-sample confidence weight w.r.t. each class k as:

$$w_{i,k} = \frac{e^{p_{i,k}}}{\sum_{b=1}^{B} e^{p_{b,k}}}, \tag{1}$$

where $p_{i,k}$ specifies the probability score that x_i belongs to the kth class. That is, $\boldsymbol{p}_i = [p_{i,1}, \dots, p_{i,K}]$, and $\sum_{k=1}^{K} p_{i,k} = 1$. Intuitively, this weight means a normalized score that a sample can represent the semantics of a class, as it is normalized across the mini-batch of samples (Fig. 2(b)).

Subsequently, we update each class prototype $\boldsymbol{c}_k$ through an efficient weighted summation operation in the feature space as:

$$\boldsymbol{c}_k = \eta \boldsymbol{c}_k + (1-\eta)\boldsymbol{c}'_k, \quad \text{with} \quad \boldsymbol{c}'_k = \sum_{i=1}^{B} w_{i,k} * \boldsymbol{f}_i \tag{2}$$

where η specifies the momentum degree for prototype update. This way, each prototype allows to improve continuously by absorbing the relevant knowledge from each individual training sample in a reliable and progressive manner during model training. This is drastically different from conventional approaches that are limited to only high-confidence samples. In contrast, our method focuses on deriving a class-level representation by extracting and summarizing knowledge

from each individual sample. Importantly, our CPD makes full use of all the training samples, whilst minimizing the notorious error propagation limitation with pseudo-labels. As a result, more comprehensive and richer knowledge about the target domain can be extracted and exploited for superior domain adaptation.

Prototype initialization: A common way to class prototype initialization is using random vectors. However, this may introduce training instability due to the cold start problem. In this work, we find that sweeping the training data D_t using the source trained model Θ_s with the proposed prototype update operation (Eq. (2)) is both more stable and better performing (see Table 5).

3.3. Training objective loss function

As each class prototype $\boldsymbol{c}_k$ is designated with a specific class label k, its ground-truth label is assigned and hence known. To form a supervision signal, we first pass it through the classifier to obtain a classification score, and then apply the standard cross-entropy loss function:

$$\mathcal{L}_{cp} = \frac{1}{K}\sum_{k=1}^{K}\mathcal{L}^{ce}(h(\boldsymbol{c}_k), k). \tag{3}$$

where $\mathcal{L}^{ce}(\cdot,\cdot)$ means cross-entropy loss function. Minimizing this loss function is separating the K class prototypes. Considering the learning process in Eq. (2), actually each prototype is a concise proxy (or delegate) of a group of semantically similar training samples. Thus, this supervision is essentially seeking for the underlying decision boundary between different classes for the target domain, which originally is ambiguous and incorrect due to the presence of cross-domain distribution shift.

Clustering target samples by class prototypes: It is found that using extra information (*e.g.* feature memory [25] and additional cluster centers [5]) is beneficial for pseudo-label estimation. In a similar spirit, we exploit class prototypes along with clustering at the beginning of each epoch. Specifically, (1) the initial cluster centers $\{\boldsymbol{cc}_k\}_{k=1}^{K}$ are initialized by class prototypes $\{\boldsymbol{c}_k\}_{k=1}^{K}$: $\boldsymbol{cc}_k = \boldsymbol{c}_k$, $k \in \{1,2,\dots,K\}$. (2) Then, each target sample x_i is labeled by the nearest cluster center $\{\boldsymbol{cc}_k\}_{k=1}^{K}$: $\hat{\boldsymbol{y}}_i = \arg\min_k cos(\boldsymbol{f}_i, \boldsymbol{cc}_k)$, where $cos(\cdot,\cdot)$ is the cosine distance function. (3) The cluster centers are further updated by these pseudo-labeled target samples: $\boldsymbol{cc}_k = \frac{1}{n_k}\sum_{i=1}^{n}\mathbb{1}_{\hat{\boldsymbol{y}}_i=k}\boldsymbol{f}_i$, $k \in \{1,2,\dots,K\}$, where $n_k = \sum_{i=1}^{n}\mathbb{1}_{\hat{\boldsymbol{y}}_i=k}$. We repeat the steps (2) and (3) until converging. All target samples finally obtain the pseudo-labels $\{\hat{\boldsymbol{y}}_i\}_{i=1}^{n}$.

Prototype regularization: Inspired by the previous UDA methods, we further introduce a prototype regularization for cross-domain distribution alignment for facilitating the domain adaptation process. To that end, existing SFDA methods [6–8,22] need to generate pseudo source samples or class prototypes. This scheme is expensive, complex, as well as error-prone due to low-quality generation. Our class prototypes can address these limitations elegantly.

By the supervision of ground-truth labels (Eq. (3)), our class prototypes are largely reliable. Even with the distribution shift, we consider that they are useful in classifying those target samples with certain similarities. With these $\boldsymbol{c}_k^i$ and pseudo-labeled target samples, we formulate a *prototype regularization* for conditional distribution alignment as:

$$\mathcal{L}_{reg} = \frac{1}{K}\sum_{k=1}^{K}\left\|\frac{1}{N_k}\sum_{i=1}^{B}\mathbb{1}_{\hat{\boldsymbol{y}}_i=k}\phi(\boldsymbol{f}_i) - \phi(\boldsymbol{c}_k)\right\|^2 \tag{4}$$

where $N_k = \sum_{i=1}^{B}\mathbb{1}_{\hat{\boldsymbol{y}}_i=k}$ is the number of target samples with pseudo-label k in the current batch, and $\phi(\cdot)$ refers to a non-linear map and $\phi(\cdot)^T\phi(\cdot)$ is a Gaussian kernel. This type of regularization has been used in conventional UDA methods [10], and we further eliminate the need for source domain training data and introduce it into SFDA.

Algorithm 1 Class Prototype Discovery for SFDA

Input: A pre-trained source model $\boldsymbol{\Theta}_s$, unlabeled training data from target domain $D_t = \{(x_i)\}_{i=1}^{n_t}$, the epoch number T, the mini-batch number M.
Output: An adapted model.
Procedure:

```
Initialize the class prototypes {c_k}_{k=1}^K with Θ_s;
for t = 1:T do
  for m = 1:M do
    Forward a mini-batch through the model;
    Compute the weights w_{i,k} (Eq. (1));
    Update {c_k}_{k=1}^K (Eq. (2));
    Calculate L_cp, L_reg, L_pl;
    Update the feature extractor with SGD (Eq. (6));
  end for
end for
return Adapted model.
```

Overall loss function: Specifically, we train the model by computing the cross-entropy between the prediction and the pseudo-label of the sample as follows:

$$\mathcal{L}_{pl} = \frac{1}{B}\sum_{i=1}^{B}\mathcal{L}^{ce}(h(g(x_i)), \hat{\boldsymbol{y}}_i). \tag{5}$$

The final objective loss of our CPD is formulated as:

$$\mathcal{L} = \mathcal{L}_{cp} + \alpha\mathcal{L}_{reg} + \beta\mathcal{L}_{pl}, \tag{6}$$

where α and β are two balancing hyper-parameters. As the training progresses, the accuracy of pseudo-labels would grow, thus increasingly beneficial. Overall, our method has two parts: class prototype discovery and exploitation. We summarize our training process in Algorithm 1.

3.4. Theoretical analysis

We make a theoretical analysis in generic domain adaptation perspective [26]. For a solution hypothesis $h \in \mathcal{H}$, the expected error for target domain $\mathcal{T}$ is generally bounded as:

$$\mathcal{R}_{\mathcal{T}}(h) \le \mathcal{R}_{\mathcal{S}}(h) + \frac{1}{2}d_{\mathcal{H}\Delta\mathcal{H}}(\mathcal{S},\mathcal{T}) + \Gamma, \tag{7}$$

where $\mathcal{R}_{\mathcal{S}}(h)$ is the expected error for domain $\mathcal{S}$. In our case, $\mathcal{S}$ is approximated by our constructed class prototypes. $d_{\mathcal{H}\Delta\mathcal{H}}(\mathcal{S},\mathcal{T})$ represents the domain discrepancy between $\mathcal{S}$ and $\mathcal{T}$, which can be minimized by our prototype regularization. Γ is the shared error across domains defined as $\Gamma = \min_{h\in\mathcal{H}}\mathcal{E}_{\mathcal{S}}(h,l_{\mathcal{S}}) + \mathcal{E}_{\mathcal{T}}(h,l_{\mathcal{T}})$ where $l_{\mathcal{S}}$ and $l_{\mathcal{T}}$ are domain-specific real labeling functions, and $\mathcal{E}_{\mathcal{X}}(\cdot)$ measures the disagreement of two labeling functions w.r.t. a specific domain $\mathcal{X}$. Under the triangle inequality [26], we have:

$$\begin{aligned}\Gamma &= \min_{h\in\mathcal{H}}\mathcal{E}_{\mathcal{S}}(h,l_{\mathcal{S}}) + \mathcal{E}_{\mathcal{T}}(h,l_{\mathcal{T}})\\ &\le \min_{h\in\mathcal{H}}\mathcal{E}_{\mathcal{S}}(h,l_{\mathcal{S}}) + \mathcal{E}_{\mathcal{T}}(h,l_{\mathcal{S}}) + \mathcal{E}_{\mathcal{T}}(l_{\mathcal{S}},l_{\mathcal{T}})\\ &\le \min_{h\in\mathcal{H}}\mathcal{E}_{\mathcal{S}}(h,l_{\mathcal{S}}) + \mathcal{E}_{\mathcal{T}}(h,l_{\mathcal{S}}) + \mathcal{E}_{\mathcal{T}}(l_{\mathcal{T}},\hat{l}_{\mathcal{T}}) + \mathcal{E}_{\mathcal{T}}(l_{\mathcal{S}},\hat{l}_{\mathcal{T}}),\end{aligned}$$

where $\hat{l}_{\mathcal{T}}$ is a pseudo-labeling function for domain $\mathcal{T}$, e.g., $h(g(\cdot))$. *First*, $\mathcal{E}_{\mathcal{S}}(h,l_{\mathcal{S}})$ and $\mathcal{E}_{\mathcal{T}}(h,l_{\mathcal{S}})$ quantify the disagreement between h and $l_{\mathcal{S}}$ on $\mathcal{S}$ and $\mathcal{T}$. By supervised training on class prototypes (Eq. (3)), h can be constrained to be close to $l_{\mathcal{S}}$ and hence restrict both disagreement to be small.

Second, $\mathcal{E}_{\mathcal{T}}(l_{\mathcal{T}},\hat{l}_{\mathcal{T}})$, the disagreement between $l_{\mathcal{T}}$ and $\hat{l}_{\mathcal{T}}$ on $\mathcal{T}$, is minimized by both the pseudo-label strategy and our $\boldsymbol{c}'_k$ (Eq. (2)) for inclining to those samples with high weights and high likelihood of

Table 1
Comparison with the state-of-the-art methods on *Office-31* dataset. Metric: classification accuracy (%); Backbone: ResNet-50; SF: Source-Free.

Method	SF	Venue	A→D	A→W	D→A	D→W	W→A	W→D	*avg*
ResNet-50 [1]	×	CVPR16	68.9	68.4	62.5	96.7	60.7	99.3	76.1
DANN [12]	×	JMLR16	79.7	82.0	68.2	96.9	67.4	99.1	82.2
DADA [27]	×	AAAI20	93.9	92.3	74.4	99.2	74.2	**100.0**	89.0
ALDA [28]	×	AAAI20	94.0	95.6	72.2	97.7	72.5	**100.0**	88.7
MDD+IA [29]	×	ICML20	92.1	90.3	75.3	98.7	74.9	99.8	88.8
ATDOC [25]	×	CVPR21	94.4	94.5	75.6	98.9	75.2	99.6	89.7
DWL [30]	×	CVPR21	91.2	89.2	73.1	**99.2**	69.8	**100.0**	87.1
CaCo [31]	×	CVPR22	81.7	89.7	73.1	98.4	72.8	**100.0**	87.6
BDCA [32]	×	AAAI22	93.8	94.0	73.5	99.0	73.0	**100.0**	88.9
ABAS [33]	×	WACV22	95.0	96.1	75.9	98.5	70.7	**100.0**	89.4
SHOT [5]	√	ICML20	94.0	90.1	74.7	98.4	74.3	99.9	88.6
3C-GAN [6]	√	CVPR20	92.7	93.7	75.3	98.5	77.8	99.8	89.6
SDDA [8]	√	WACV21	85.3	82.5	66.4	99.0	67.7	99.8	83.5
SFIT [7]	√	CVPR21	89.9	91.8	73.9	98.7	72.0	99.9	87.7
A^{2}Net [34]	√	ICCV21	94.5	94.0	76.7	**99.2**	76.1	**100.0**	90.1
SFDA-DE [22]	√	CVPR22	96.0	94.2	76.6	98.5	75.5	99.8	90.1
DIPE [35]	√	CVPR22	**96.6**	93.1	75.5	98.4	77.2	99.6	90.1
D-MCD [23]	√	AAAI22	94.1	93.5	76.4	98.8	76.4	**100.0**	89.9
CPD	√	Ours	**96.6** ± **0.1**	94.2 ± 0.4	**77.3** ± **0.2**	98.2 ± 0.1	**78.3** ± **0.3**	**100.0** ± **0.0**	**90.8**

being correctly predicted. Specifically, for the kth prototype of the target domain, it is calculated as,

$$\boldsymbol{c}_k = \sum_{i=1}^{n_t} w_{ik} \boldsymbol{f}_i^t \approx \sum_{\boldsymbol{f}_i^t \in set_k^t} w_{ik} \boldsymbol{f}_i^t, \tag{8}$$

where set_k^t is a set of target samples with high weights w_{ik}, that is, $\hat{l}_{\mathcal{T}}(\boldsymbol{f}_i^t) = k$ when $\boldsymbol{f}_i^t \in set_k^t$. It means the kth prototype $\boldsymbol{c}_k$ is an approximation of the features from the kth category. When these prototypes are optimized using classification loss (Eq. (3)), $h(\boldsymbol{c}_k)$ will get more closer to the real label k, essentially making the feature in set_k^t close to the real label k. Therefore $\mathcal{E}_{\mathcal{T}}(l_{\mathcal{T}}, \hat{l}_{\mathcal{T}})$ is minimized.

Third, $\mathcal{E}_{\mathcal{T}}(l_S, \hat{l}_{\mathcal{T}})$ is constrained by our prototype regularization, which essentially pull pseudo-labeled samples to the corresponding class prototype $\boldsymbol{c}_k$. Specifically, when we perform semantic alignment of the kth category, the prototype regularization optimizes the following equation,

$$\left\| \frac{1}{N_k} \sum_{\boldsymbol{f}_i^t \in set_k^t} \phi(\boldsymbol{f}_i^t) - \phi(\boldsymbol{c}_k) \right\|^2, \tag{9}$$

By optimizing Eq. (9), the feature $\boldsymbol{f}_i^t \in set_k^t$ will be close to the kth prototype $\boldsymbol{c}_k$, which makes $l_S(\boldsymbol{f}_i^t) = l_S(\boldsymbol{c}_k) = k$. So $\mathcal{E}_{\mathcal{T}}(l_S, \hat{l}_{\mathcal{T}})$ is minimized.

In summary, the overall upper bounds will be decreased. This theoretically proves that our method works.

4. Experiments

In this section, we first describe our experimental setup, which includes an introduction to the dataset, implementation details, and other competitors. Then we present the comparison results with other state-of-the-art methods. Finally, we further set up some ablation experiments to further verify the effectiveness of our method.

4.1. Experimental setup

Datasets: Like most domain adaptation methods, there are three standard datasets we choose, namely ***Office-31***, ***Office-Home***, ***Visda-17***.

Office-31 [36][2] is a standard benchmark which contains 4,652 images from 31 office environment categories. It is collected from three different domains: Amazon (A) downloaded from online website, Webcam (W) and DSLR (D) taken by web camera and digital SLR camera respectively.

Office-Home [37][3] is a more challenging dataset with 15,588 images from 65 classes in four domains: Artistic images (Ar), Clip-Art images (Cl), Product images (Pr) and RealWorld images (Rw).

Visda-17 [38][4] is another challenging domain adaptation benchmark for synthesis-to-real object recognition task. It has 12 classes, including a source domain with 152,397 synthetic images and a target domain with 55,388 real object images.

Implementation details: Our experiments are implemented on Pytorch platform. For each experiment, the same setting is run for five times to enhance robustness and report the mean and standard deviation. For fair comparison with existing methods, we use the same CNN backbone (Resnet50 for Office-31 and office-home and Resnet101 for Visda-17). The bottleneck network and classifier both consist of one fully-connected layers in all cases. We use the SGD optimizer with the cosine learning rate scheduler. We set empirically the hyper-parameters as follows: $\eta = 0.8$ for Eq. (2), $\alpha = 0.3$ for all datasets, $\beta = 0.1/0.1/0.3$ for Office-31/Office-Home/Visda-17 in Eq. (6).

Competitors: We compare three groups of alternative methods from the literature. **(1)** *Direct Transfer* (DT) of a supervised learning model (e.g., ResNet [1]) trained on the source domain without any adaptation towards the target domain. This is useful to reveal the domain shift problem. **(2)** Previous state-of-the-art Unsupervised Domain Adaptation (UDA) methods that require both domain's training data for domain adaptation. These include DANN [12], DADA [27], ALDA [28], MDD+IA [29], ATDOC [25] and DWL [30], CaCo [31], BDCA [32], ABAS [33]. **(3)** Recent state-of-the-art Source-Free Domain Adaptation (SFDA) methods that need no access to the source domain data for domain adaptation. They are SHOT [5], 3C-GAN [6], SDDA [8], SFIT [7], DIPE [35], D-MCD [23], SFDA-DE [22], A^{2}Net [34] and G-SFDA [39].

4.2. Result analysis

Results on Office-31: The results of our CPD and other state-of-the-art methods are reported in Table 1. Out of all six tasks, our CPD produces the best results in 4 of them. In the remaining two tasks, our CPD does not lag far behind the best competitor. Especially in task W→A, CPD

[2] http://www.eecs.berkeley.edu/~mfritz/domainadaptation/

[3] https://www.hemanthdv.org/officeHomeDataset.html

[4] http://ai.bu.edu/visda-2017/

Table 2
Comparisons with the state-of-the-art methods on *Office-Home* dataset. Metric: classification accuracy (%); Backbone: ResNet-50; SF: Source-Free.

Method	SF	Ar→Cl	Ar→Pr	Ar→Rw	Cl→Ar	Cl→Pr	Cl→Rw	Pr→Ar	Pr→Cl	Pr→Rw	Rw→Ar	Rw→Cl	Rw→Pr	*avg*
ResNet-50[1]	×	34.9	50.0	58.0	37.4	41.9	46.2	38.5	31.2	60.4	53.9	41.2	59.9	46.1
DANN [12]	×	45.6	59.3	70.1	47.0	58.5	60.9	46.1	43.7	68.5	63.2	51.8	76.8	57.6
CDAN+E [2]	×	50.7	70.6	76.0	57.6	70.0	70.0	57.4	50.9	77.3	70.9	56.7	81.6	65.8
ABAS [33]	×	51.5	71.7	75.5	59.8	69.4	69.5	59.8	47.1	77.7	70.6	55.2	80.2	65.7
MDD+IA [29]	×	56.2	77.9	79.2	64.4	73.1	74.4	64.2	54.2	79.9	71.2	58.1	83.1	69.5
ATDOC [25]	×	58.3	78.8	82.3	**69.4**	78.2	78.2	67.1	56.0	82.7	72.0	58.2	**85.5**	72.2
BDCA [32]	×	51.8	73.9	80.7	66.3	71.8	74.2	65.3	51.8	81.1	**74.7**	58.5	84.6	69.6
SHOT [5]	√	57.1	78.1	81.5	68.0	78.2	78.1	67.4	54.9	82.2	73.3	58.8	84.3	71.8
G-SFDA [39]	√	57.9	78.6	81.0	66.7	77.2	77.2	65.6	56.0	82.2	72.0	57.8	83.4	71.3
DIPE [35]	√	56.5	**79.2**	80.7	70.1	**79.8**	78.8	**67.9**	55.1	**83.5**	74.1	59.3	84.8	72.5
D-MCD [23]	√	**59.4**	78.9	80.2	67.2	79.3	78.6	65.3	55.6	82.2	73.3	**62.8**	83.9	72.2
CPD	√	59.1	79.0	**82.4**	68.5	79.7	**79.5**	**67.9**	**57.9**	82.8	73.8	61.2	84.6	**73.0**
		±0.2	±0.2	±0.1	±0.1	±0.1	**±0.1**	±0.2	**±0.1**	±0.1	±0.1	±0.1	±0.1	

Table 3
Comparison with the state-of-the-art methods on *Visda-17* dataset. Metric: per-class classification accuracy (%); Backbone: ResNet-101; SF: Source-Free.

Method	SF	plane	bcycl	bus	car	horse	knife	mcycl	person	plant	sktbrd	train	truck	*avg*
ResNet-101 [1]	×	55.1	53.3	61.9	59.1	80.6	17.9	79.7	31.2	81.0	26.5	73.5	8.5	52.4
DANN [12]	×	81.9	77.7	82.8	44.3	81.2	29.5	65.1	28.6	51.9	54.6	82.8	7.8	57.4
CDAN [2]	×	85.2	66.9	83.0	50.8	84.2	74.9	88.1	74.5	83.4	76.0	81.9	38.0	73.9
DADA [27]	×	92.9	74.2	82.5	65.0	90.9	93.8	87.2	74.2	89.9	71.5	86.5	48.7	79.8
ALDA [28]	×	93.8	74.1	82.4	69.4	90.6	87.2	**89.0**	67.6	93.4	76.1	87.7	22.2	77.8
DWL [30]	×	90.7	80.2	**86.1**	67.6	92.4	81.5	86.8	78.0	90.6	57.1	85.6	28.7	77.1
ATDOC [25]	×	93.7	83.0	76.9	58.7	89.7	95.1	84.4	71.4	89.4	80.0	86.7	55.1	80.3
CaCo [31]	×	90.4	80.7	78.8	57.0	88.9	87.0	81.3	79.4	88.7	88.1	86.8	**63.9**	80.9
SHOT [5]	√	94.3	**88.5**	80.1	57.3	93.1	94.9	80.7	80.3	91.5	89.1	86.3	58.2	82.9
3C-GAN [6]	√	94.8	73.4	68.8	**74.8**	93.1	95.4	88.6	**84.7**	89.1	84.7	83.5	48.1	81.6
SFIT [7]	√	94.3	79.0	84.9	63.6	92.6	92.0	88.4	79.1	92.2	79.8	87.6	43.0	81.4
A^2Net [34]	√	94.0	87.8	85.6	66.8	93.7	95.1	85.8	81.2	91.6	88.2	86.5	56.0	84.3
DIPE [35]	√	95.2	87.6	78.8	55.9	93.9	95.0	84.1	81.7	92.1	88.9	85.4	58.0	83.1
CPD	√	**96.7**	**88.5**	79.6	69.0	**95.9**	**96.3**	87.3	83.3	94.4	**92.9**	87.0	58.7	**85.8**

leads the best competitors by 1.9%. Compared with ATDOC, which is a best competitor in the UDA setting, CPD still exceeds it by 1.1% in a more challenging SFDA setting. Compared with A^2Net, SFDA-DA and DIPE, which are three best competitors in the SFDA setting, CPD still surpasses these methods 0.7%, which proves the effectiveness of our method.

Results on Office-Home: As shown in Table 2, on this more challenging test benchmark we have largely similar observations as above. For example, our CPD remains the best performer, whilst other SFDA methods expect some recent methods fail to catch up the best UDA model ATDOC. Importantly, the margin of the best over the remaining is further enlarged. This indicates the stability and generalization of our method.

Results on Visda-17: This presents a different test scenario on the transfer between synthetic and real image data. As shown in Table 3, this is not an exceptional case in comparisons between best UDA and SFDA methods. Again, our CPD method remains the best adaption method among all the alternatives, although on a number of tasks different methods may excel. Encouragingly, we see that CPD can outperform the best UDA method, BCDM, by a margin of 2.2%. This consistently shows the stable performance advantage of our method over all the existing competitors, verifying the efficacy of our class prototype discovery idea.

From Tables 1–3, we can make the following observations. **(1)** The domain shift in data distribution indeed challenges the source domain's model (Reset50 in this case), leading to significant inferior model generalization. **(2)** Compared with the UDA setting, SFDA is more challenging. So it is not surprising to see that most SFDA methods are outperformed by best UDA models. However, we also see some exceptions including the very recent methods, and our CPD. This is a phenomenal achievement as the conventional wisdom would consider the best UDA performance as an upper bound for SFDA. This suggests that source data is not necessary at all provided that the pre-trained model and target domain data both can be well exploited. **(3)** Among the top-3 methods, our CPD is the best on the most transfer tasks as well as in the overall performance. This clearly demonstrates the superiority of our method over the state-of-the-art alternatives despite its simplicity.

4.3. Ablation study

In this section, we further construct some experiments to analyze our method and prove its effectiveness. First, we conducted a component analysis to analyze the effectiveness of each part of our method. Then, the learned features of different combinations are visualized by t-SNE to give an intuitive understanding. Next, the hyperparameters α, β, and η are analyzed to demonstrate the robustness of our CPD. Next, we analyzed how to initialize prototype. Finally, we compare class prototype with current pseudo label strategy to further verify the effectiveness of discovered class prototype.

Component analysis: We conduct component analysis including class prototype (CP), prototype regularization (PR), and pseudo-labels (PL) based on ***Office-31*** datasets. Since the accuracies on the W→D and D→W transfer tasks are relatively high, so we perform other four task experiments to show the performances of our method with different parts.

Table 4 shows that (1) the class prototypes alone can provide strong results across all the tasks, laying down a solid performance ground. (2) When then class prototypes are further used for domain adaptation, a clear accuracy gain is achieved. This suggests the importance of conditional distribution alignment thanks to a reliable usage of our class prototypes. (3) A further boost is realized from integrating the

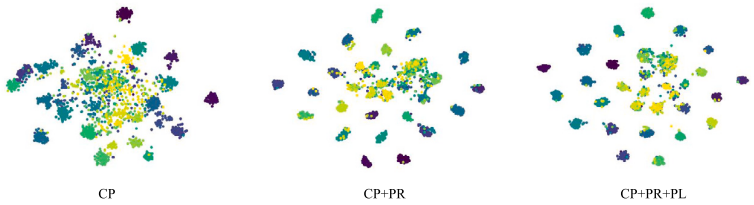

Fig. 3. Component analysis by T-SNE based feature visualization on the task D→A of $Office-31$. CP: Class Prototype; PR: Prototype Regularization; PL: Pseudo-label.

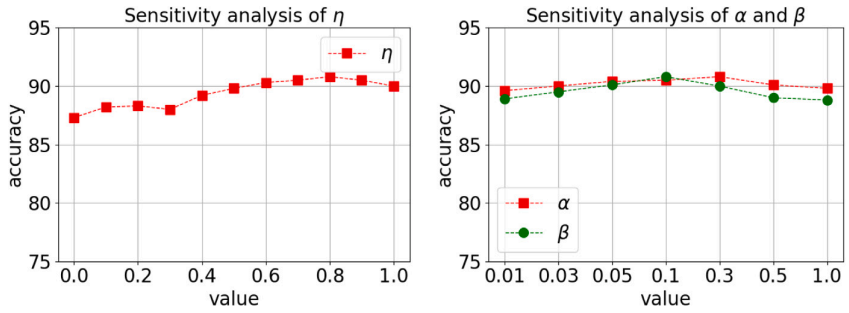

Fig. 4. Sensitivity analysis of (l) moving average momentum η and (r) loss weights α and β on $Office-31$.

Table 4
Component analysis on *Office-31*. CP: Class Prototype; PR: Prototype Regularization; PL: Pseudo-label.

Component	A→D	A→W	D→A	W→A
Direct transfer	68.9 ± 0.1	68.4 ± 0.1	62.5 ± 0.1	60.7 ± 0.1
CP	91.2 ± 0.1	90.1 ± 0.2	73.4 ± 0.1	73.9 ± 0.1
CP+PR	94.9 ± 0.2	93.4 ± 0.3	76.4 ± 0.1	77.4 ± 0.1
CP+PR+PL	**96.6 ± 0.1**	**94.2 ± 0.4**	**77.3 ± 0.2**	**78.3 ± 0.3**

pseudo-labels, suggesting a good compatibility of our method with the self-training pipeline.

Visualization by t-SNE: To give an intuitive understanding of our method, the target features on the transfer task D→A are further visualized by t-SNE [40] in Fig. 3. The features in Fig. 3(a) are learned by the class prototype discovery. The features shown in Fig. 3(b) are extracted by network g optimized by both class prototype discovery and the prototype regularization. Fig. 3(c) shows the features by minimizing CPD strategy, prototype regularization and pseudo-label strategy simultaneously. It can be observed that the features in Fig. 3(a) are very scattered. Compared with Fig. 3(a), the features in Fig. 3(b) are more concentrated due to the influence of prototype regularization. The features are more concentrated in Fig. 3(c) since pseudo-label strategy is further added. From left to right, the features are obtained by optimizing the class prototype discovery, both prototype regularization and class prototype discovery, and all terms. And the experimental results are consistent with the component analysis.

Parameter analysis: To verify the robustness of our method, we further analyze the sensitivity of parameters η, α and β in Eqs. (2) and (6), respectively. Their experimental results on all transfer tasks of Office-31 are shown in Fig. 4.

For the parameter η, the result is shown in Fig. 4(a). On the whole, the experimental performance is increased first and then decreased with the increase of η. To achieve the better performance, η is set as 0.8 for trading off in all experiments.

For the parameters α and β, the results are shown in Fig. 4(b). Similar to η, the experimental performance is increased first and then decreased with the increase of α and β. In our method, $\alpha = 0.3$ for all datasets, $\beta = 0.1/0.1/0.3$ for Office-31/Office-Home/Visda-17.

Table 5
Effect of class prototype initialization on *Office-31*.

Init.	A→D	A→W	D→A	W→A
Random	93.9 ± 0.4	92.8 ± 0.2	76.3 ± 0.4	77.1 ± 0.3
Ours	**96.6 ± 0.1**	**94.2 ± 0.4**	**77.3 ± 0.2**	**78.3 ± 0.3**

Table 6
The comparison between our class prototype (CP) discovery and conventional pseudo-label (PL) strategy on *Office-31*.

Component	A→D	A→W	D→A	W→A
PL	84.2 ± 0.2	83.2 ± 0.3	68.9 ± 0.2	66.9 ± 0.2
CP	**91.2 ± 0.1**	**90.1 ± 0.2**	**73.4 ± 0.1**	**73.9 ± 0.1**

From results, we can find that the performance have little effect on the changes of all hyperparameters, which shows the robustness of our method.

Prototype initialization: To examine the effect of class prototype initialization, we further compare our sweeping method (Section 3.2) with the *random* vector baseline.

Table 5 shows that using the source model for prototype initialization is consistently superior. In contrast, the baseline will suffer a cold start which harms the performance.

Class prototype vs. pseudo-label: We further compare the performance of our class prototype (CP) idea with the conventional pseudo-label (PL) strategy. Both CP and PL leverage classifier's predictions but differ in some key manners. PL uses classifier's predictions to obtain pseudo-labels for self-training. In contrast, CP aggregates the classifier's predictions with predictive confidences over *all the training samples* to construct a class prototype for each class on which we impose the learning constraints. This sets our method apart from PL which however imposes learning supervision on *training samples individually* without tolerance to predictive uncertainty and mistakes. We report the results on top-4 challenging tasks of ***Office-31*** are shown in Table 6. It is evident that our CP is clearly superior over PL in making use of class predictions for SFDA.

Noise analysis of pseudo-labels: We assess the noise degree of pseudo-labels by our method (*i.e.* class prototypes based K-means clustering per

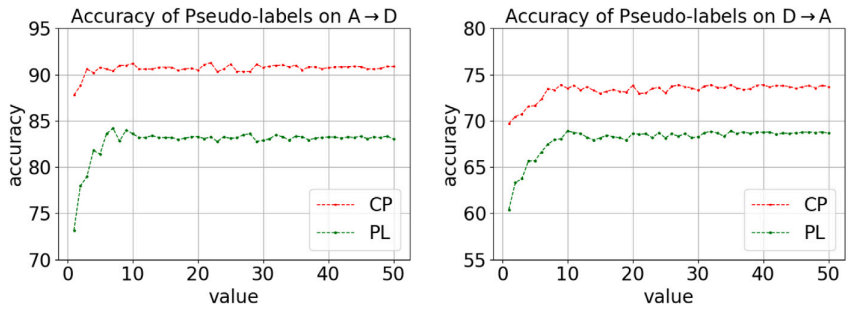

Fig. 5. Pseudo-label noise analysis on A→D and D→A.

epoch) and a baseline that uses the network predictions. Fig. 5 shows that our method is more reliable, especially in the beginning of training.

4.4. Limitation and future work

In our experiments, it is assumed that the source and target domains share the same set of classes (i.e., the closed-set setting). In many real-world situations, however, this might be not true but unknown/novel classes may appear in the target domain. Therefore, how to extend our current model by integrating the ability of discovering novel classes would be useful, which is part of our future work.

5. Conclusions

We have proposed a novel *class prototype discovery* (CPD) method for source-free domain adaption. It is built on the introduction of semantic class prototypes. To discover the optimal prototypes more reliably, we design a classification score-based learning mechanism to enable a full usage of all the target domain training data. Further, we introduce a prototype regularization for conditional distribution alignment via class prototypes. CPD is not only concise and effective, but also complementary to conventional self-training with theoretical justification. Finally, we conduct experiments on three public datasets to verify our CPD in comparison to a wide range of alternative methods, along with in-depth ablation study and visualization. Although achieving new state-of-the-art performance, our evaluation assumes the closed-set setting where all the classes are aligned across the source and target domains. In practice, there could be new classes involved in the training data. Extending our method to the more realistic open-set setting would be another significant research in the future.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

The authors do not have permission to share data.

Acknowledgment

This work was supported by National Natural Science Foundation of China (62276048).

References

[1] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 770–778.
[2] M. Long, Z. Cao, J. Wang, M.I. Jordan, Conditional adversarial domain adaptation, in: NeurIPS, 2018.
[3] L. Zhou, M. Ye, S. Xiao, Domain adaptation based on source category prototypes, Neural Comput. Appl. 34 (23) (2022) 21191–21203.
[4] L. Zhou, M. Ye, D. Zhang, C. Zhu, L. Ji, Prototype-based multisource domain adaptation, IEEE Trans. Neural Netw. Learn. Syst. 33 (10) (2021) 5308–5320.
[5] J. Liang, D. Hu, J. Feng, Do we really need to access the source data? Source hypothesis transfer for unsupervised domain adaptation, in: International Conference on Machine Learning, PMLR, 2020, pp. 6028–6039.
[6] R. Li, Q. Jiao, W. Cao, H.-S. Wong, S. Wu, Model adaptation: Unsupervised domain adaptation without source data, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 9641–9650.
[7] Y. Hou, L. Zheng, Visualizing adapted knowledge in domain transfer, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021, pp. 13824–13833.
[8] V.K. Kurmi, V.K. Subramanian, V.P. Namboodiri, Domain impression: A source data free domain adaptation method, in: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, 2021, pp. 615–625.
[9] W. Li, S. Chen, Unsupervised domain adaptation with progressive adaptation of subspaces, Pattern Recognit. 132 (2022) 108918.
[10] P. Ge, C.-X. Ren, X.-L. Xu, H. Yan, Unsupervised domain adaptation via deep conditional adaptation network, Pattern Recognit. 134 (2023) 109088.
[11] X.-L. Xu, G.-X. Xu, C.-X. Ren, D.-Q. Dai, H. Yan, Conditional independence induced unsupervised domain adaptation, Pattern Recognit. (2023) 109787.
[12] Y. Ganin, E. Ustinova, H. Ajakan, P. Germain, H. Larochelle, F. Laviolette, M. Marchand, V. Lempitsky, Domain-adversarial training of neural networks, J. Mach. Learn. Res. 17 (1) (2016) 2096–2030.
[13] Q. He, Q. Dai, X. Wu, J.-Y. He, A novel class restriction loss for unsupervised domain adaptation, Neurocomputing 461 (2021) 254–265.
[14] M.-Y. Liu, O. Tuzel, Coupled generative adversarial networks, in: Proceedings of the 30th International Conference on Neural Information Processing Systems, 2016, pp. 469–477.
[15] K. Saito, K. Watanabe, Y. Ushiku, T. Harada, Maximum classifier discrepancy for unsupervised domain adaptation, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 3723–3732.
[16] L. Zhou, M. Ye, X. Zhu, S. Li, Y. Liu, Class discriminative adversarial learning for unsupervised domain adaptation, in: Proceedings of the 30th ACM International Conference on Multimedia, 2022, pp. 4318–4326.
[17] B.B. Damodaran, B. Kellenberger, R. Flamary, D. Tuia, N. Courty, Deepjdot: Deep joint distribution optimal transport for unsupervised domain adaptation, in: Proceedings of the European Conference on Computer Vision, ECCV, 2018, pp. 447–463.
[18] Y. Du, D. Zhou, Y. Xie, Y. Lei, J. Shi, Prototype-guided feature learning for unsupervised domain adaptation, Pattern Recognit. 135 (2023) 109154.
[19] Q. He, Z. Yuan, X. Wu, J.-Y. He, Domain-specific conditional jigsaw adaptation for enhancing transferability and discriminability, in: Proceedings of the 30th ACM International Conference on Multimedia, 2022, pp. 6327–6336.
[20] G. French, M. Mackiewicz, M. Fisher, Self-ensembling for visual domain adaptation, in: International Conference on Learning Representations, no. 6, 2018.
[21] L. Zhou, S. Xiao, M. Ye, X. Zhu, S. Li, Adaptive mutual learning for unsupervised domain adaptation, IEEE Trans. Circuits Syst. Video Technol. (2023).
[22] N. Ding, Y. Xu, Y. Tang, C. Xu, Y. Wang, D. Tao, Source-free domain adaptation via distribution estimation, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022, pp. 7212–7222.
[23] T. Chu, Y. Liu, J. Deng, W. Li, L. Duan, Denoised maximum classifier discrepancy for sourcefree unsupervised domain adaptation, in: Thirty-Sixth AAAI Conference on Artificial Intelligence, AAAI-22, vol. 2, 2022.

[24] L. Xiong, M. Ye, D. Zhang, Y. Gan, Y. Liu, Source data-free domain adaptation for a faster R-CNN, Pattern Recognit. 124 (2022) 108436.
[25] J. Liang, D. Hu, J. Feng, Domain adaptation with auxiliary target domain-oriented classifier, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021, pp. 16632–16642.
[26] S. Ben-David, J. Blitzer, K. Crammer, A. Kulesza, F. Pereira, J.W. Vaughan, A theory of learning from different domains, Mach. Learn. 79 (1) (2010) 151–175.
[27] H. Tang, K. Jia, Discriminative adversarial domain adaptation, in: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 34, no. 04, 2020, pp. 5940–5947.
[28] M. Chen, S. Zhao, H. Liu, D. Cai, Adversarial-learned loss for domain adaptation, in: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 34, no. 04, 2020, pp. 3521–3528.
[29] X. Jiang, Q. Lao, S. Matwin, M. Havaei, Implicit class-conditioned domain alignment for unsupervised domain adaptation, in: International Conference on Machine Learning, PMLR, 2020, pp. 4816–4827.
[30] N. Xiao, L. Zhang, Dynamic weighted learning for unsupervised domain adaptation, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021, pp. 15242–15251.
[31] J. Huang, D. Guan, A. Xiao, S. Lu, L. Shao, Category contrast for unsupervised domain adaptation in visual tasks, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022, pp. 1203–1214.
[32] G. Yang, M. Ding, Y. Zhang, Bi-directional class-wise adversaries for unsupervised domain adaptation, Appl. Intell. 52 (4) (2022) 3623–3639.
[33] L. Robbiano, M.R.U. Rahman, F. Galasso, B. Caputo, F.M. Carlucci, Adversarial branch architecture search for unsupervised domain adaptation, in: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, 2022, pp. 2918–2928.
[34] H. Xia, H. Zhao, Z. Ding, Adaptive adversarial network for source-free domain adaptation, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2021, pp. 9010–9019.
[35] F. Wang, Z. Han, Y. Gong, Y. Yin, Exploring domain-invariant parameters for source free domain adaptation, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022, pp. 7151–7160.
[36] K. Saenko, B. Kulis, M. Fritz, T. Darrell, Adapting visual category models to new domains, in: European Conference on Computer Vision, Springer, 2010, pp. 213–226.
[37] H. Venkateswara, J. Eusebio, S. Chakraborty, S. Panchanathan, Deep hashing network for unsupervised domain adaptation, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 5018–5027.
[38] X. Peng, B. Usman, N. Kaushik, J. Hoffman, D. Wang, K. Saenko, Visda: The visual domain adaptation challenge, 2017, arXiv preprint arXiv:1710.06924.
[39] S. Yang, Y. Wang, J. van de Weijer, L. Herranz, S. Jui, Generalized source-free domain adaptation, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2021, pp. 8978–8987.
[40] L.v.d. Maaten, G. Hinton, Visualizing data using t-SNE, J. Mach. Learn. Res. 9 (Nov) (2008) 2579–2605.

Lihua Zhou received the B.S. degree in Internet of Things engineering from the Hefei University of Technology, Xuancheng, China, in 2019. He is currently pursuing the Ph.D. degree with the University of Electronic Science and Technology of China, Chengdu, China. His current research interests include machine learning, computer vision, and transfer learning.

Nianxin Li received the B.S. degree in Computer Science and Technology from the University of Electronic Science and Technology of China, Chengdu, China in 2022. He is currently pursuing the M.S. degree with the University of Electronic Science and Technology of China, Chengdu, China. His current research interests include machine learning, computer vision, and transfer learning.

Mao Ye received the B.S. degree from Sichuan Normal University in mathematics, Chengdu, China, in 1995, the M.S degree in mathematics from the University of Electronic Science and Technology of China, Chengdu, in 1998, and the Ph.D. degree in mathematics from the Chinese University of Hong Kong, China, in 2002. He has been a short-time Visiting Scholar with the University of Queensland and the University of Pennsylvania. He is currently a Professor and the Director of CVLab with the University of Electronic Science and Technology of China, Chengdu. His research interests include machine learning and computer vision. In these areas, he has published over 90 papers in leading international journals or conference proceedings. He has served on the editorial board of Engineering Applications of Artificial Intelligence. He was a Co-Recipient of the Best Student Paper Award at the IEEE ICME 2017.

Xiatian Zhu received the Ph.D. from the Queen Mary University of London. He won The Sullivan Doctoral Thesis Prize 2016, an annual award representing the best doctoral thesis submitted to a U.K. University in computer vision. His research interests include computer vision and machine learning.

Song Tang received the B.S. degree in electronic engineering from Hainan University, Haikou, China, in 2005, the M.S. degree in system engineering from the Chongqing University of Posts and Communications, Chongqing, China, in 2013, and the Ph.D. degree in computer science from the University of Electronic Science and Technology of China, Chengdu, China, in 2017. From 2017 to 2019, he was an Assistant Researcher with the Department of Informatik, Universit at Hamburg, Hamburg, Germany. He is currently an Associate Professor with the Institute of Machine Intelligence, University of Shanghai for Science and Technology. He has authored or coauthored more than 30 journal articles. His current research interests include transfer learning, reinforcement learning, and computer vision. Dr. Tang was a recipient of the International Association of Geomagnetism and Aeronomy Young Scientist Award for Excellence, in 2008, and the IEEE Electromagnetic Compatibility Society Best Symposium Paper Award, in 2011.